



Sparse approximation based resource allocation in DSL/DMT transceivers with per-tone equalization



Prabin Kumar Pandey^{a,b,*}, Marc Moonen^{a,b}, Luc Deneire^c

^a Department of Electrical Engineering, KU Leuven, Belgium

^b IBBT Future Health Department, Belgium

^c Laboratoire I3S, University of Nice, Sophia-Antipolis, France

ARTICLE INFO

Article history:

Received 17 October 2012

Received in revised form

19 July 2013

Accepted 25 July 2013

Available online 15 August 2013

Keywords:

Multicarrier communication

DMT

DSL

PTEQ

Resource allocation

Sparse approximation

ABSTRACT

Per-tone equalization has been proposed as an alternative to time domain equalization for DMT receivers in DSL modems. It optimizes the bit rate performance of the receiver as each tone can be equalized independently. It has also been shown that using variable length equalizers can significantly reduce the total number of equalizer taps and hence the run-time complexity, without compromising performance. For a given transmit power loading, it has been shown that the equalizer taps can be allocated optimally using a dual decomposition based approach with per-tone exhaustive searches over all possible equalizer lengths. However, a more general approach is needed when optimal transmit power allocation is also considered to maximize the overall bit rate, where in addition the per-tone exhaustive searches are replaced by a more efficient procedure. In this paper, a sparse approximation based resource allocation algorithm is presented to allocate equalizer taps and transmit power over tones and maximize the overall bit rate. This algorithm is shown to provide efficient allocations at a relatively low computational cost.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Digital Subscriber Line (DSL) modems use Discrete Multi-Tone (DMT) modulation. DMT divides the available spectrum into smaller parallel sub-bands or tones. Each tone corresponds to an orthogonal carrier. In the transmitter, the input bit-stream is divided into several independent parallel streams which then QAM-modulate the different carriers. These QAM symbols are then inputs to an inverse discrete Fourier transform (IDFT) block. A cyclic prefix is added to each resulting time domain symbol before transmission, which allows for an easy channel equalization at the receiver. However, if the cyclic prefix is shorter than the channel impulse response, this results in inter-symbol

interference (ISI) and inter-carrier interference (ICI). Highly dispersive channels such as the ADSL channel have a very long channel impulse response hence to mitigate ISI/ICI a very long cyclic prefix is needed. As a long cyclic prefix results in a large transmission overhead, channel equalization is used in the receiver to shorten the effective channel impulse response [1,2]. The usual time domain equalization (TEQ) [3] corresponds to a joint equalization of all the tones and cannot optimize the performance in each and every tone. An alternative frequency domain equalization technique, known as per-tone equalization (PTEQ), has been proposed in [4] in order to equalize each tone separately. It is then possible to optimize the bit rate performance of the receiver using an optimal equalizer for every tone, even without increasing the overall run-time computational complexity.

In a PTEQ based DMT receiver, typically every tone is equalized using a constant length (T-taps) equalizer. However, the transmission channel gain varies for different

* Corresponding author at: Department of Electrical Engineering, KU Leuven, Belgium. Tel.: +32 16321924.

E-mail addresses: pkumarpa@esat.kuleuven.be, prabin.pandey@gmail.com (P.K. Pandey).

tones, and so for a tone with a low channel gain using a long equalizer does not increase its bit rate performance significantly. Therefore, using a constant length equalizer for all tones may unnecessarily increase the run-time complexity and correspond to a waste of system resources. For a given transmit power loading, an efficient algorithm to distribute a given equalizer tap budget over tones has been presented in [5]. This algorithm is based on a dual problem formulation and involves a per-tone exhaustive search over all possible equalizer lengths. However, when optimal transmit power allocation is also considered to maximize the overall bit rate along with the equalizer tap allocation, then the exhaustive search would be over all possible power levels and all possible equalizer lengths for each tone, and so would become prohibitive. Hence an alternative method of allocating resources over tones is needed.

For a given equalizer tap budget and total transmit power budget, the overall bit rate maximization problem can be written as a dual optimization problem using Lagrange multipliers. In this paper, an algorithm to determine the optimal equalizer tap and transmit power allocation over the used tones is proposed using sparse approximation. This algorithm is shown to provide efficient allocations at a relatively low computational cost. There has been various greedy algorithms as well as convex relaxation based approach to sparse filter design in the literature [9–11].

In Section 2, the basic data model is provided and per-tone equalization is reviewed. In Section 3, the resource allocation problem is formulated. An algorithm to solve this problem using sparse approximation is developed in Sections 4 and 5. Section 6 contains some simulation results. Finally conclusion are presented in Section 7.

2. Preliminaries

2.1. Data model

The following notation is adopted in the description of the DMT system: N is the size of the (I)DFT and ν represents the length of the cyclic prefix, $s = N + \nu$, i and k denote the tone index and DMT symbol index respectively, \mathcal{F}_N and \mathcal{I}_N are the N -point DFT and IDFT matrices where $\mathcal{F}_N(i, :)$ is the i th row of \mathcal{F}_N , T is the maximum equalizer length and the equalizer coefficients vector for tone i is \mathbf{v}_i , \mathbf{I}_Q and $\mathbf{0}_Q$ are the $Q \times Q$ identity and zero matrices respectively, $X_i^{(k)}$ is a complex subsymbol on tone i ($i = 1 \dots N$) in DMT symbol k , $X_{1:N}^{(k)} = [X_1^{(k)} \dots X_N^{(k)}]^T$, $y_{\tilde{k}}$ represents the received signal and $n_{\tilde{k}}$ represents the additive noise at time \tilde{k} , $\mathbf{h} = [h_L \dots h_0 \dots h_{-K}]$ is the channel impulse response in reverse order, $\{\cdot\}^T$ denotes the transpose, $\{\cdot\}^*$ denotes the conjugate.

The received signal can be modeled as

$$\begin{bmatrix} y_{ks+\nu-T+2+\delta} \\ \vdots \\ y_{(k+1)s+\delta} \end{bmatrix} \uparrow N+T-1$$

$$= \begin{bmatrix} \mathbf{h} & \mathbf{0} & \dots \\ \ddots & \ddots & \ddots \\ \mathbf{0} & \dots & \mathbf{h} \end{bmatrix} \begin{bmatrix} \mathbf{P} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathcal{I}_N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{n} \end{bmatrix} + \begin{bmatrix} X_{1:N}^{(k-1)} \\ X_{1:N}^{(k)} \\ X_{1:N}^{(k+1)} \end{bmatrix} + \begin{bmatrix} n_{ks+\nu-T+2+\delta} \\ \vdots \\ n_{(k+1)s+\delta} \end{bmatrix} \quad (1)$$

here δ is the synchronization delay and is a design parameter. Matrix \mathbf{P} adds the cyclic prefix and is given as

$$\mathbf{P} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_\nu \\ \mathbf{I}_N & \mathbf{0} \end{bmatrix}.$$

In (1), the k th symbol is the symbol of interest, the $(k-1)$ th and the $(k+1)$ th symbol have been used to fully describe the ISI. For further details on the used data model we refer to [4].

2.2. Per-tone equalization

Per-tone equalization (PTEQ) can be derived from time domain equalization (TEQ) as follows [4]. First, the TEQ-based receiver operation can be specified as

$$Z_i^{(k)} = D_i \mathcal{F}_N(i, :)(\mathbf{Y}\mathbf{w}), \quad (2)$$

which is equivalent to

$$Z_i^{(k)} = \text{row}_i(\underbrace{\mathcal{F}_N \mathbf{Y}}_{T \text{ FFTs}}) \underbrace{\mathbf{w} D_i}_{T\text{-tap PTEQ}}, \quad (3)$$

where row_i operator represents the i -th row of the matrix within the operator, \mathbf{w} is the vector representing the T -tap time domain equalizer, D_i is the single tap frequency domain equalizer coefficient for tone i , $Z_i^{(k)}$ is the equalizer output for tone i and symbol k and \mathbf{Y} is the $N \times T$ Toeplitz matrix given as

$$\mathbf{Y} = \begin{bmatrix} y_{ks+\nu+1} & y_{ks+\nu} & \dots & y_{ks+\nu-T+2} \\ y_{ks+\nu+2} & y_{ks+\nu+1} & \dots & y_{ks+\nu-T+3} \\ \ddots & \ddots & \ddots & \ddots \\ y_{(k+1)s} & y_{(k+1)s-1} & \dots & y_{(k+1)s+\nu-T+1} \end{bmatrix},$$

where $\delta = 0$ for conciseness.

In (3), it can be seen that T DFT operations are needed to equalize one symbol compared to one DFT operation per symbol when the usual TEQ operation (2) is used. However, it has been shown in [4] that (3) can be written in terms of a sliding DFT and then eventually one DFT and $T-1$ difference terms are needed to equalize one symbol, i.e.

$$Z_i^{(k)} = \mathbf{w}^T D_i \begin{bmatrix} \mathcal{F}_N(i, :) & \mathbf{0} & \dots \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathcal{F}_N(i, :) \end{bmatrix} \mathbf{y} \quad (4)$$

where \mathcal{I} is the set containing the indices of the used tones, $b_i = \log_2(1 + \text{SNR}_i/\Gamma)$ is the number of bits that can be loaded on tone i , SNR_i is the output SNR obtained for tone i for a given equalizer \mathbf{v}_i as given by (7) and Γ is the so-called SNR gap, $C_{ij} \in \{0, 1\}$, C_{ij} is equal to 1 if the j -th equalizer tap for tone i is selected to be non-zero and 0 otherwise, C_{budget} is the predefined maximum total number of non-zero equalizer taps, $\mathbf{c}_i^T = [C_{i1} \dots C_{iT}]$, \mathbf{C} is a matrix that has \mathbf{c}_i^T as its i -th row, $s_i \in \mathcal{S}$ is the transmit power on tone i , \mathcal{S} is the set containing all possible discrete transmit power levels, \mathbf{s} is a vector that has s_i as its i -th row, S_{budget} is the maximum total transmit power, $s_{i,\text{max}}$ is the maximum transmit power allowed on tone i . Note that \mathbf{c}_i defines the zero/non-zero taps (i.e. the sparsity pattern) of \mathbf{v}_i . The MMSE-PTEQ filter design formula (6) is easily modified to take such a sparsity pattern into account.

This primal optimization problem is coupled over tones and is a combinatorial problem. It has a computational complexity of $\mathcal{O}(L^M 2^{MT})$, where L is the cardinality of the set \mathcal{S} , i.e. the total number of discrete transmit power levels and M is the cardinality of set \mathcal{I} , i.e. the total number of used tones. This is intractable even for moderate values of L , M and T . In [5], it has been shown that for fixed transmit powers, (8) decouples over tones when formulated as a dual optimization problem, thus reducing the computational complexity. Here the dual problem formulation of (8) can be written as

$$\begin{aligned} & \underset{\lambda, \gamma}{\text{minimize}} \left\{ \underset{\mathbf{C}, \mathbf{s}}{\text{maximize}}(\mathcal{L}) \right\} \\ & \text{where } \mathcal{L} = \sum_{i \in \mathcal{I}} b_i + \lambda \left(C_{\text{budget}} - \sum_{i \in \mathcal{I}} \sum_{j=1}^T C_{ij} \right) + \gamma \left(S_{\text{budget}} - \sum_{i \in \mathcal{I}} s_i \right), \end{aligned} \quad (9)$$

where λ and γ are known as the Lagrange multipliers and \mathcal{L} is the Lagrangian. For given values of λ and γ the maximization in (9) will be replaced by

For $i = 1 \dots M$

$$\begin{aligned} & \underset{\mathbf{c}_i, s_i}{\text{maximize}}(\mathcal{L}_i) \quad \text{where } \mathcal{L}_i = b_i - \lambda \sum_{j=1}^T C_{ij} - \gamma s_i, \\ & \text{end} \end{aligned} \quad (10)$$

where \mathcal{L}_i is the per-tone Lagrangian. Note that b_i in (10) is a function of \mathbf{s} (not just s_i) and so an iterative maximization may indeed be needed. Even when (10) is decoupled over tones, an exhaustive search has to be performed over all possible values of s_i and all possible vectors \mathbf{c}_i . For given λ and γ , the computational complexity is $\mathcal{O}(ML2^T)$, which is still large for large T . It can be seen that if the sparsity patterns of the equalizers can somehow be controlled then the combinatorial search is avoided and the exponential complexity is reduced to a linear complexity in T .

In [5], the sparsity pattern is restricted to combinations where only contiguous taps can be non-zero thus reducing the computational complexity from $\mathcal{O}(ML2^T)$ to $\mathcal{O}(MLT)$. This will be referred to as the contiguous tap selection (CTS) approach. Thus the combinatorial search is reduced to a linear search in the MMSE-PTEQ filter orders. However, restricting the sparsity pattern may not be the best approach, since the best sparsity pattern may not be in the restricted

search space. In the following section a sparse approximation based approach will be used to design a sparse PTEQ filter.

4. Sparse approximation based equalizer design

In order to find a better way to control the sparsity pattern, the MMSE-PTEQ design problem (6) can be written as a sparse approximation problem [6–8]

$$\underset{\mathbf{v}_i}{\text{minimize}} \quad \|\mathbf{A}_i \mathbf{v}_i^* - \tilde{\mathbf{b}}_i\|_2^2 + \tau \|\mathbf{v}_i\|_0, \quad (11)$$

where $\|\cdot\|_0$ is the ℓ_0 quasi-norm of the vector in the argument, i.e., it counts the number of non-zero elements of the vector in the argument, τ controls the trade-off between the sparsity and the quadratic term. Problem (11) is however known to be NP hard in general [6]. To simplify (11) the non-convex ℓ_0 quasi-norm is often replaced by the convex ℓ_1 norm [6–8]. This can be written as

$$\underset{\mathbf{v}_i}{\text{minimize}} \quad \|\mathbf{A}_i \mathbf{v}_i^* - \tilde{\mathbf{b}}_i\|_2^2 + \beta \|\mathbf{v}_i\|_1, \quad (12)$$

where now β controls the trade-off between the sparsity and the quadratic term. Eq. (12) is a convex problem and can be solved using any generic solver in polynomial time [12]. If the underlying system admits a sparse solution, it has been shown that solving (12) is equivalent to solving (11) [7, 8]. In this case, however, the underlying system does not necessarily admit a sparse solution, therefore a sparse MMSE-PTEQ filter can not be obtained by just solving (12). One way to obtain a sparse filter is to adopt a two step procedure. Firstly, (12) can be used, for a given β , to obtain a nearly-sparse solution and then the coefficients below a certain threshold level ζ are forced to zero. Secondly, the sparsity pattern thus obtained can be used to compute the corresponding MMSE-PTEQ filter using (6).

It is clear that the choice of the trade-off parameter β and ζ is important for the algorithm to work properly. An efficient update rule for the β is based on the difference between the total available system resources and the used system resources for the current value of β . The difference in the total available system resource and the used system resource provides the direction of the correction for β . This can be written as

$$\beta^{t+1} = \left[\beta^t - \mu \left(C_{\text{budget}} - \sum_{i \in \mathcal{I}} \|\mathbf{v}_i\|_0 \right) \right]^+ \quad (13)$$

where β^t is the value of β at t -th iteration, μ is a step size parameter, and $[a]^+$ is $\max(0, a)$. For simplicity, in the simulations μ is set to a fixed arbitrary value close to zero. The threshold level can be fixed to a constant level or can also be updated iteratively to speed up the convergence. It is obvious that if the threshold level is set to a higher value, the probability of a sparser equalizer becomes higher. The update formula for the threshold level can therefore be written as

$$\zeta^{t+1} = \left[\zeta^t - \sigma \left(C_{\text{budget}} - \sum_{i \in \mathcal{I}} \|\mathbf{v}_i\|_0 \right) \right]^+, \quad (14)$$

where σ is a step size parameter.

In (14), the threshold is same for all the tones. However, the MMSE-PTEQ filters for different tones generally yield different MMSEs. The lower MMSE tones can then have a

higher threshold than higher MMSE tones. This can be achieved by turning (14) into a per-tone threshold level update formula, i.e.

$$\zeta_i^{t+1} = \left[\zeta_i^t - \sigma_i \left(C_{\text{budget}} - \sum_{i \in \mathcal{I}} \|\mathbf{v}_i\|_0 \right) \right]^+, \quad (15)$$

where

$$\sigma_i = \|\mathbf{A}_i \mathbf{v}_i^* - \mathbf{b}_i\|_2^2 \quad (16)$$

5. Sparse approximation based resource allocation

With $\|\mathbf{v}_i\|_0 = \sum_{j=1}^T C_{ij}$, (10) can also be written as maximize $b_i - \lambda \|\mathbf{v}_i\|_0 - \gamma s_i$. (17)

Now for each discrete power level s_i , a sparse \mathbf{v}_i can be computed using the method described in Section 4. This results in solving (12) and (6) once for each power level. Even though the equalizer coefficients are computed twice, for large T this has a similar computational complexity as the CTS approach of [5], which requires solving (6) T times for each power level. The computational complexity can be further reduced by solving (12) only once for the initial power level and then using the same sparsity pattern for all the other power levels.

The Lagrange multiplier γ , which enforces the transmit power constraint, has to be updated based on the difference between the current total transmit power and the total transmit power budget, i.e.

$$\gamma^{t+1} = \left[\gamma^t - \eta \left(S_{\text{tot}} - \sum_{i \in \mathcal{I}} s_i \right) \right]^+, \quad (18)$$

where η is a step size parameter. Similarly, to enforce the total tap constraint the Lagrange multiplier λ can be updated as

$$\lambda^{t+1} = \left[\lambda^t - \theta \left(C_{\text{budget}} - \sum_{i \in \mathcal{I}} \|\mathbf{v}_i\|_0 \right) \right]^+, \quad (19)$$

where θ is a step size parameter.

An algorithm to allocate the resources, i.e. the equalizer taps and the transmit power, for given resource constraints is given in Algorithm 1.

Algorithm 1. Sparse approximation based resource allocation.

- 1: Initialize vector containing transmit power \mathbf{s} , s_i^{opt}
- 2: Initialize Lagrange multipliers λ and γ
- 3: Initialize trade-off parameter β and threshold ζ_i , $i \in \mathcal{I}$
- 4: Initialize step sizes parameters μ , θ , and η
- 5: **repeat**
- 6: **for** tone $i \in \mathcal{I}$ **do**
- 7: Compute \mathbf{v}_i using (12) and \mathbf{s}
- 8: Find the sparsity pattern \mathbf{c}_i by applying the threshold to \mathbf{v}_i
- 9: Initialize $\mathcal{L}_i^{\text{opt}} = 0$
- 10: **for** power level $s_i \in \mathcal{S}$ **do**
- 11: Compute \mathbf{v}_i with sparsity pattern \mathbf{c}_i using (6)
- 12: Compute the objective function \mathcal{L}_i using (17)
- 13: **if** $\mathcal{L}_i \geq \mathcal{L}_i^{\text{opt}}$ **then**
- 14: $s_i^{\text{opt}} \leftarrow s_i$
- 15: $\mathcal{L}_i^{\text{opt}} \leftarrow \mathcal{L}_i$

```

16:   end if
17: end for
18: Replace transmit power tone  $i$  in  $\mathbf{s}$  with  $s_i^{\text{opt}}$ 
19:  $\mathbf{v}_i^{\text{opt}} \leftarrow \mathbf{v}_i$ 
20: end for
21: Update  $\beta$ ,  $\gamma$ ,  $\zeta_i$  and  $\lambda$  using (13), (18), (15), and (19)
22: until  $(S_{\text{budget}} - \sum_{i \in \mathcal{I}} s_i \leq \text{tolerance} \ \& \ C_{\text{budget}} - \sum_{i \in \mathcal{I}} \|\mathbf{v}_i\|_0 \leq \text{tolerance})$ 

```

6. Simulation results

The simulation results presented here refer to an ADSL scenario. The same setting as in [5] was used in order to be able to compare the results. In these simulations, the synchronization delay δ was not considered. Simulations were performed on a standardized ADSL channel model (CSA loop) [13]. The bit error rate probability was fixed to 10^{-7} , the coding gain and the noise margin were 3 dB and 6 dB respectively. The signal and noise PSD (power spectral density) levels were -40 dB and -140 dB respectively and the total transmit power budget was 100 mW. Three consecutive symbols were considered to account for the ISI. The channel was assumed to be known perfectly at the receiver. The first 38 tones were not used. The maximum number of equalizer taps T per tone was 20. To compute the sparsity pattern using (12), any available numerical solver can be used e.g. CVX [14,15]. Then the required sparse PTEQ can be computed with (6) using a least squares method.

Figs. 2 and 3 show the bit rates versus the total number of equalizer taps for two different cases. In the first case, Fig. 2, there is no transmit power loading in order to compare the results with those of [5]. From Fig. 2, it can be seen that for a given bit rate the sparse approximation based equalizer resource allocation method always performs better than the CTS approach of [5]. Therefore we can conclude that for the same computational complexity

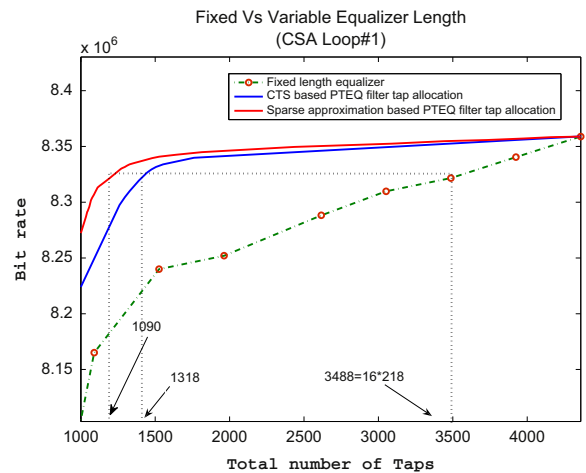


Fig. 2. Comparison between the total number of equalizer taps needed to achieve the same bit rate performance for fixed length equalizer and variable length equalizer in CSA loop 1 without any power loading (for the fixed length equalizer, the total number of taps is 218* number of taps per tone).

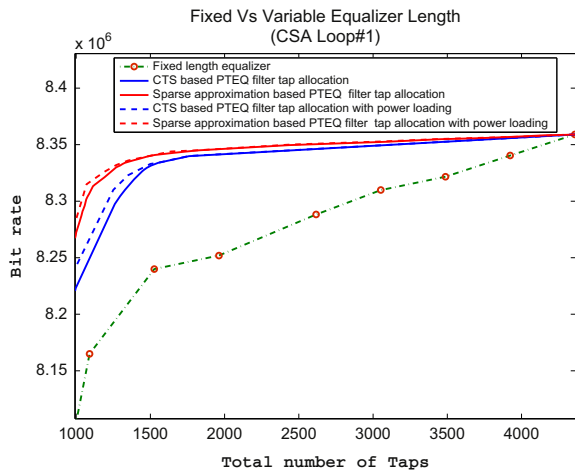


Fig. 3. Comparison between the performance of the sparse MMSE-PTEQ with and without power loading.

a better performance can always be obtained using the sparse approximation based resource allocation.

In the second case, Fig. 3, a total transmit power constraint is also enforced. The total transmit power constraint is 100 mW. In this case there are 218 used tones and -40 dBm maximum PSD. We can see from Fig. 3 that the sparse approximation based MMSE-PTEQ filter tap allocation combined with power loading performs better.

7. Conclusion

In this paper, an improved resource allocation method using sparse approximation has been presented for PTEQ-based DSL/DMT transceivers. For a fixed power loading, the proposed method computes the sparsity patterns of the MMSE-PTEQ filters such that a global equalizer tap constraint is satisfied, and is shown to provide a better solution than [5] for similar computational complexity. With combined transmit power and MMSE-PTEQ filter tap allocation, it was shown that a better performance is achieved at significantly reduced complexity.

Acknowledgments

This research work was carried out at the ESAT Laboratory of KU Leuven, Leuven, Belgium, in the frame of KU Leuven Research Council CoE EF/05/006 Optimization in Engineering (OPTEC), PFV/10/002 (OPTEC), the Belgian Programme on Interuniversity Attraction Poles initiated

by the Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, “Dynamical systems, control and optimization”, 2007–2011), Research Project FWO nr. G.0235.07 Design and evaluation of DSL systems with common mode signal exploitation, Marie-Curie EST-SIGNAL program (<http://est-signal.i3s.unice.fr>) under contract No. MEST-CT-2005-021175, IWT Project “PHANTER: Physical layer and Access Node TEchnology Revolutions: enabling the next generation broadband network”, and Concerted Research Action GOA-MaNet. The scientific responsibility is assumed by its authors.

References

- [1] J. Cioffi, P. Silverman, T. Starr, *Understanding Digital Subscriber Line Technology*, 1st ed. Prentice Hall, 1999.
- [2] J. Bingham, Multicarrier modulation for data transmission: an idea whose time has come, *IEEE Communications Magazine* 28 (May) (1990) 5–14.
- [3] N. Al-Dahir, J.M. Cioffi, Optimum finite-length equalization for multicarrier transceivers, *IEEE Transactions on Communications* 44 (January) (1996) 56–64.
- [4] K.V. Acker, G. Leus, M. Moonen, O.V. de Wiel, T. Pollet, Per tone equalization for DMT-based systems, *IEEE Transactions on Communications* 49 (January) (2001) 109–119.
- [5] P.K. Pandey, M. Moonen, Resource allocation in ADSL variable length per-tone equalizers, *IEEE Transactions on Signal Processing* 56 (May) (2008) 2161–2164.
- [6] J.A. Tropp, Just relax: convex programming methods for subset selection and sparse approximation, ICES Report 04-04, February 2004, pp. 1–39.
- [7] J.A. Tropp, Just relax: convex programming methods for identifying sparse signals in noise, *IEEE Transactions on Information Theory* 52 (March) (2006) 1030–1050.
- [8] D.L. Donoho, M. Elad, V.N. Temlyakov, Stable recovery of sparse overcomplete representations in the presence of noise, *IEEE Transactions on Information Theory* 52 (January) (2006) 6–18.
- [9] A. Gama, N. Al-Dahir, A new design framework for sparse FIR MIMO equalizers, *IEEE Transactions on Communications* 59 (August) (2011) 2132–2140.
- [10] T. Baran, D. Wei, A.V. Oppenheim, Linear programming algorithms for sparse filter design, *IEEE Transactions on Signal Processing* 58 (March) (2010) 1605–1617.
- [11] D. Needell, J.A. Tropp, COSAMP: iteratives signal recovery from incomplete and inaccurate samples, *Communications of the ACM* 53 (January) (2010).
- [12] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, NY, USA, 2004.
- [13] ANSI, Network and customer installation interfaces—ADSL metallic interface, in: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, Technical Report, May 2006.
- [14] M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 1.21, (<http://cvxr.com/cvx>), October 2010.
- [15] M. Grant, S. Boyd, Graph implementations for nonsmooth convex programs, in: V. Blondel, S. Boyd, H. Kimura (Eds.), *Recent Advances in Learning and Control*, Series: Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008, pp. 95–110. (<http://stanford.edu/~boyd/graph-dcp.html>).